Protein structure optimization with a "Lamarckian" ant colony algorithm

Oakley, Mark T; Richardson, E Grace; Carr, Harriet; Johnston, Roy L

DOI:
10.1109/TCBB.2013.125

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Oakley, MT, Richardson, EG, Carr, H & Johnston, RL 2013, 'Protein structure optimization with a "Lamarckian" ant colony algorithm', *IEEE - ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 6, pp. 1548-52. https://doi.org/10.1109/TCBB.2013.125

Protein Structure Optimisation With a "Lamarckian" Ant Colony Algorithm

Mark T. Oakley, E. Grace Richardson, Harriet Carr, Roy L. Johnston *†

July 6, 2014

Abstract

We describe the LamarckiAnt algorithm: a search algorithm that combines the features of a "Lamarckian" genetic algorithm and ant colony optimisation. We have implemented this algorithm for optimisation of BLN model proteins, which have frustrated energy landscapes and represent a challenge for global optimisation algorithms. We demonstrate that LamarckiAnt performs competitively with other state-of-the-art optimisation algorithms.

1 Introduction

Locating the global minimum structure of a flexible molecule can be a difficult problem. Even relatively small molecules can have sufficient degrees of freedom to make finding the global minimum by exhaustive searches computationally infeasible. Larger molecules, such as peptides and proteins, have hundreds or thousands of degrees of freedom and present a difficult optimisation challenge. Numerous global optimisation algorithms have been applied to the problem of locating the most stable structures on the potential energy surface. By far the

*The authors acknowledge the Engineering and Physical Sciences Research Council (EPSRC) for funding (Programme Grant EP/I001352/1).

†The authors are with the School of Chemistry, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK. Email: m.t.oakley@bham.ac.uk, r.l.johnston@bham.ac.uk

most widely used methods are those based on Metropolis Monte Carlo [1–6] or genetic algorithms [5–10]. Other algorithms, such as particle swarm optimisation [11, 12], immune algorithms [13] and artificial bee colonies [14] have also been used.

One search algorithm that has only seen limited applications in molecular structure optimisation [15–18] is ant colony optimisation (ACO) [19]. This algorithm is inspired by the foraging of colonies of ants, which tend to find the shortest path to a source of food in spite of the fact that individual ants have no knowledge of the overall landscape. As ants forage, they lay down a trail of pheromone that slowly evaporates. Other ants tend to follow more intense pheromone trails, which leads to shorter paths being reinforced by repeated visits from several ants while longer paths dissipate. Eventually, most of the pheromone remains on the shortest available path. Originally, the ACO algorithm was applied to the travelling salesman problem (TSP) [19].

The ACO algorithm must be modified for use in structure optimisation problems, requiring a representation of the structure that can be treated as an ant's path and a way of assigning the energy of the structure to the length of the path. The optimisation of protein structures on a regular lattice is a discrete problem like TSP, and a protein structure can simply be expressed as a walk over this lattice. The length of the path is related to the energy of the structure, with more stable structures corresponding to shorter paths. ACO gives comparable performance to other state-of-the-art algorithms for the optimisation of lattice proteins [15, 16].

To optimise chemical structures that are not constrained to a lattice, ACO must be further modified to deal with optimisation of continuous functions. Daeyaert et al. [17] optimised the structures of a series of small molecules, where each structure was represented by a series of torsion angles corresponding to the freely-rotatable bonds in the molecule. The torsion angles were discretised into bins, and one of these discrete values was selected for each angle. To account for the continuous nature of the energy function, pheromone was laid down in the

chosen bin, with smaller amounts deposited in neighbouring bins. Dresselhaus et al. modified this approach by using particle-swarm optimisation to optimise the parameters used in the ACO search [18].

An important development in the global optimisation of chemical structures is the basin-hopping (BH) principle of performing a local minimisation of all candidate structures generated by a search algorithm. Local minimisation transforms the potential energy surface into a series of steps, each of which is the basin of attraction of a minimum [20]. This removes the downhill barriers between minima and allows larger moves over the potential energy surface to be attempted. In the BH variant of Monte Carlo optimisation, structures are subjected to random deformation and then locally minimised before performing a Metropolis acceptance test [2–6]. Similarly, "Lamarckian" genetic algorithms (GA) locally optimise all offspring and mutants before selecting the individuals that go forward to the next generation [5, 6, 8]. Here, we describe an ACO algorithm that includes a local optimisation stage. We call this method LamarckiAnt because it is an ACO algorithm that includes some of the features of a "Lamarckian" GA. We test the performance of the LamarckiAnt algorithm on two BLN model proteins, which are difficult challenges for optimisation algorithms because of their frustrated energy landscapes.

2 Methods

2.1 BLN proteins

The BLN-model is a coarse-grained potential for modelling proteins [21, 22]. It treats a protein as a string of beads, with one bead per peptide residue. The beads are divided into three classes: hydropho**B**ic, hydrophi**L**ic and **N**eutral. We use two BLN-model proteins as test systems for the LamarckiAnt algorithm (Fig 1). The 46-residue protein, $B_9N_3(LB)_4N_3B_9N_3(LB)_5L$, folds into a four-strand β-barrel and has a frustrated energy landscape, with several competing low-energy structures separated by high barriers [5, 6, 21–24]. The 69-residue

protein, $B_9N_3(LB)_4N_3B_9N_3(LB)_4N_3B_9N_3(LB)_5L$, forms a six-strand β-barrel and has a more frustrated energy landscape [5, 6, 25–28]. These will be referred to as BLN-46 and BLN-69 in the rest of this manuscript. The largest molecule previously investigated with ACO was a peptide with 10 rotatable torsion angles [18]. The BLN-model proteins have 43 and 66 rotatable torsions and therefore have substantially larger conformational spaces.

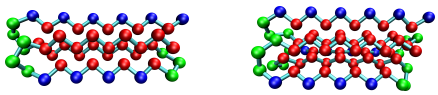

Figure 1: Global minima of the BLN-46 (left) and BLN-69 (right) model proteins.

For consistency with previous studies [5, 6, 23, 24, 29], the energies of the protein structures were evaluated using a version of the BLN potential with bond lengths and angles constrained by stiff spring constants [22]:

$$\begin{aligned} V_{BLN} =& \frac{1}{2}K_r \sum_{i=1}^{N-1}(R_{i,i+1} - R_e)^2 + \frac{1}{2}K_\theta \sum_{i=1}^{N-2}(\theta_i - \theta_e)^2 \\ &+ \epsilon \sum_{i=1}^{N-3}\left[A_i(1+\cos\phi_i) + B_i(1+3\cos\phi_i)\right] \\ &+ 4\epsilon \sum_{i=1}^{N-2}\sum_{j=i+2}^{N} C_{ij}\left[\left(\frac{\sigma}{R_{ij}}\right)^{12} - D_{ij}\left(\frac{\sigma}{R_{ij}}\right)^{6}\right], \end{aligned} \tag{1}$$

where R_{ij} is the distance between atoms i and j, θ_i is the angle between atoms $i, i+1$ and $i+2$ and ϕ_i is the torsion angle given by atoms $i, i+1, i+2$ and $i+3$. The first two terms are stiff harmonic angle and bond restraints with $K_r = 231.2\,\epsilon\sigma^{-2}$, $R_e = \sigma$, $K_\theta = 20$ rad^{-2} and $\theta_e = 1.8326$ rad. In the third term $A = B = 1.2$ unless two or more of the beads involved are N. In this case, $A = 0$ and $B = 0.2$. The parameters for the non-bonded terms are listed in Table 1.

	B	L	N
B	$C = 1$, $D = 1$	$C = 2/3$, $D = -1$	$C = 1$, $D = 0$
L	$C = 2/3$, $D = -1$	$C = 2/3$, $D = -1$	$C = 1$, $D = 0$
N	$C = 1$, $D = 0$	$C = 1$, $D = 0$	$C = 1$, $D = 0$

Table 1: Parameters used in the non-bonded term of the BLN potential.

2.2 LamarckiAnt Algorithm

The LamarckiAnt algorithm is based on Daeyaert's ACO [17], but with some modifications (Fig 2). In each cycle of the algorithm, m ants are generated, each of which encodes a single conformation of the protein. Each ant's route comprises the sequence of torsion angles in the backbone of the protein chain. The torsion angle space is divided into bins of width $\Delta\phi$. Each torsion angle, ϕ_i in an ant's route is assigned by roulette selection with the size of each sector of the roulette wheel given by:

$$p(\phi_i, t) = \tau(\phi_i, t) \tag{2}$$

where $\tau(\phi_i, t)$ is the amount of pheromone present for a given residue and torsion angle at iteration t. The values of the torsion angles within each bin are then assigned randomly. The values of $\tau(\phi_i, t)$ are always normalised. In the first iteration of the algorithm, $\tau(\phi_i, 0)$ is a uniform probability function.

In Daeyaert's ACO, the selection of torsion angles includes an additional factor based on the torsional energy term from the force field [17]. This biases the search towards torsion angles that are locally stable, but requires the implementation of a system-dependent term in the optimisation algorithm. The LamarckiAnt algorithm does not include this term and the selection of torsional angles is based only on the pheromone trail. After the random torsion angles are assigned, each structure is then relaxed to the nearest local minimum using the limited-memory Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS) [30]

algorithm as implemented in GMIN [31]. The values of ϕ_i are then replaced with those from the corresponding optimised structure and the pheromone trails are generated from these updated routes. Local minimisation has been shown to be very effective in combination with other optimisation algorithms [3, 8] and has the additional advantage of not requiring the implementation of a system-specific torsional biasing term.

After m ants have made their walks, the pheromone trails are updated. The amount of pheromone laid down by ant k is determined by the learning rate, Q^k, which is given by:

$$Q^k = e^{-\gamma(e_k - e_{min})} \tag{3}$$

where e_k is the energy associated with ant k and e_{min} is the lowest energy found so far. Thus, low energy structures result in more pheromone being deposited, with the parameter γ determining the weighting for less-stable solutions.

The trail update, $\Delta\tau(\phi_i)$, for each bin for a given torsion angle is given by:

$$\Delta\tau(\phi_i) = N \sum_k \frac{Q^k}{\sqrt{2}\pi w} e^{\frac{-(\phi_i - \phi_i^k)^2}{2w^2}}. \tag{4}$$

The parameter, w, determines the width of the pheromone trail laid down by each ant. The normalisation constant, N, ensures that the sum of all the probabilities for a single torsion angle is unity. The pheromone trails are then updated as:

$$\tau(\phi_i, t+1) = \rho\tau(\phi_i, t) + (1-\rho)\Delta\tau(\phi_i) \tag{5}$$

where ρ is the persistence of the pheromone trail and takes a value between 0 and 1.

```
initialise pheromone trails
while (not converged)
    for k = 1, ..., n
        ant k makes walk
        relax structure corresponding to ant k
        replace ant k's path with minimised path
    end loop over n
    calculate Q_k for each ant from relaxed structure
    if no improvement for m cycles
        re-set pheromone trails
    else
        update pheromone trails
    end if
end cycle
```

Figure 2: Pseudocode describing the LamarckiAnt algorithm.

Some additional modifications to Daeyaert's ACO [17] were tested. We employ a restart operator in the LamarckiAnt algorithm that is analogous to the epoch operator in our GA [5]. If the energy of the lowest structure found since the last restart does not improve for a number of cycles, m, the pheromone trail is re-set to a uniform probability distribution. Using a restart operator ensures that, eventually, all searches locate the global minimum. In ACO searches on other system [18, 32], the efficiency of the search is improved by including the best solution found so far in the trail update. We include a global best update in the LamarckiAnt algorithm, with a fraction, g, of the trail update in each cycle supplied by the best solution found so far. Note that the global best update is taken from the best solution found in the whole search, which allows some information about the search to be retained after a restart. There are seven parameters that can influence the performance of the LamarckiAnt algorithm.

We study the effect of varying g and m in this manuscript. The values of the other parameters are kept constant and are shown in Table 2.

Parameter		Value
Number of ants	n	50
Learning rate constant	γ	2.5
Pheromone persistence	ρ	0.1
Pheromone width	w	20°
Bin width	$\Delta\phi$	10°
Restart length	m	50

Table 2: Parameters used for LamarckiAnt optimisation of the BLN model proteins.

The BLN-46 and BLN-69 proteins have been studied extensively, and the global minimum structures for both are known. We measured the performance of the LamarckiAnt algorithm by recording the mean time to the first encounter of the global minimum from 100 independent searches. All searches were allowed to run until they located the known global minimum of BLN-46 or BLN-69. We quote these times in terms of the number of minimisation operations and the number of energy evaluations, both of which are independent of the computer hardware used in the calculations. To the best of our knowledge, the fastest published optimisations of the BLN proteins have been obtained using the BH and GA approaches [5], and we compare LamarckiAnt to these. The published times for the GA were obtained by using single-point crossover in the mating step [5]. We have subsequently found that two-point crossover improves the efficiency of the GA, and these results are also presented here.

3 Results

When optimising BLN-46, the best published performance [5] was obtained by the BH algorithm, which required an average of 4400 minimisations to locate

the global minimum (Table 3). GAs using one- or two-point crossover are both slower than this in terms of the number of minimisations required to locate the global minimum. The LamarckiAnt algorithm requires an average of only 2600 minimisations to find the global minimum. The mean first encounter time on a single core of a 2.2 GHz Intel Sandy Bridge E5-2660 processor is 100 s. The use of a restart operator is unnecessary for BLN-46, because all searches locate the global minimum rapidly. The best performance of the LamarckiAnt algorithm is obtained when the global best structure is not included in the trail update ($g = 0$). Larger values of g lead to an increase in the time required to locate the global minimum.

	Mean first encounter time	
Method	Energy evaluations	Minimisations
BH [5]	6.7×10^5 (5.6×10^5)	4.4×10^3 (3.8×10^3)
GA-1pt [5]	1.4×10^6 (9.3×10^5)	8.3×10^3 (5.7×10^3)
GA-2pt	1.2×10^6 (7.0×10^5)	5.1×10^3 (3.4×10^3)
LamarckiAnt		
$g = 0$	1.3×10^6 (8.5×10^5)	2.6×10^3 (1.7×10^3)
$g = 0.25$	1.6×10^6 (1.3×10^6)	3.2×10^3 (2.7×10^3)
$g = 0.5$	2.6×10^6 (7.5×10^6)	6.0×10^3 (1.7×10^4)
$g = 0.75$	3.1×10^6 (6.6×10^6)	5.8×10^3 (6.7×10^3)

Table 3: Mean first encounter times for 100 global optimisation runs from random starting positions of the BLN-46 protein. Values in parentheses are the standard deviations.

To demonstrate the importance of the local optimisation step, optimisation of BLN-46 was attempted using Daeyaert's original ACO algorithm [17]. A series of 100 ACO runs were performed, each of which proceeded was allowed to run for 10^5 cycles, giving 5×10^6 energy evaluations in each run. The most stable structures found were $> 50\epsilon$ above the global minimum with four extended strands that had not folded into a β-barrel. The global minimum has

a densely-packed structure and structures that deviate from it by a small distance can have very high energies due to the r^{-12} repulsive term in the BLN potential (1). Without local minimisation, these structures make a negligible contribution to the pheromone trail. Gradient-driven minimisation removes the overlapping residues and the resulting structures make a larger contribution to the pheromone trail.

Analysis of the ants' paths (Fig 3) and the pheromone trails (Fig 4) shows how the LamarckiAnt algorithm locates the global minimum. Here, we analyse the search with the median first encounter time from the searches where $g = 0$. In the initial population, solutions with a range of values for all 43 of the torsion angles are present. As the search proceeds, the extended conformations of the four strands of the β-barrel are located at different times, with the $(LB)_5L$ strand at the C-terminus (residues 36-46) found first and the B_9 strand at the N-terminus (residues 1-9) found last. The correct conformations of the turn residues are not located until very late in the search.

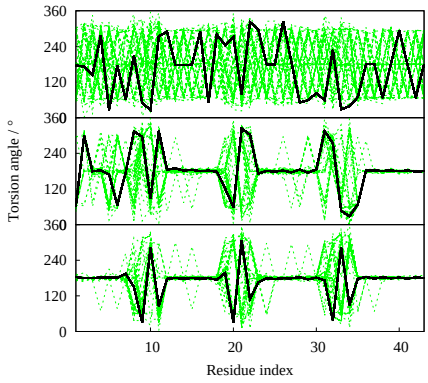

Figure 3: Optimised paths taken by the ants in the first (top), tenth (middle) and final (bottom) cycles of a typical run of the LamarckiAnt algorithm on BLN-46. The most stable structure (solid line) and all of the other structures in the current cycle (dotted lines) are displayed. In the final cycle, the most stable structure is the known global minimum. The index is of the first residue of the torsion angle $i, \ldots i + 3$.

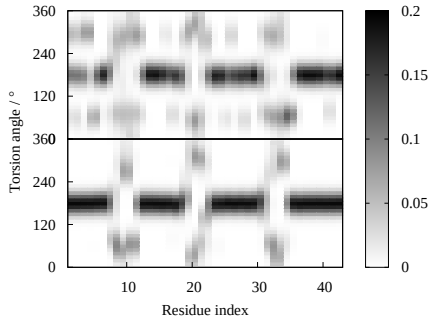

Figure 4: Intensity of the pheromone trail, τ, before the tenth (top) and final (bottom) cycles of a typical LamarckiAnt optimisation of BLN-46. The index is of the first residue of the torsion angle $i, \ldots i+3$.

The published optimisation times [5] for BH and the GA with one-point crossover on BLN-69 are very similar (Table 3). Here, the use of two-point crossover in the GA gives a significant improvement in the efficiency of the search. The performance of the LamarckiAnt algorithm is less competitive for this system. Without the use of a global best update, the search rapidly locates structures $\sim 5\epsilon$ above the global minimum. From there, it visits several solutions of similar energy, but only rarely finds structures that are more stable. The use of a global best update improves the mean first encounter time, with $g = 0.75$ making the searches three times faster compared to $g = 0$. However, LamarckiAnt is still slower than BH or the GA when optimising BLN-69.

The restart operator plays an important role in allowing the LamarckiAnt algorithm escape from traps. With $m = 50$, the searches require an average of 17 restarts to find the global minimum. In the range $10 \leq m \leq 100$, the performance is not very sensitive to the value of m. The mean first encounter time on a single core of a 2.2 GHz Intel Sandy Bridge E5-2660 processor is 5600 s.

Method	Mean first encounter time	
	Energy evaluations	Minimisations
BH [5]	4.8×10^6 (4.0×10^6)	2.6×10^4 (2.3×10^4)
GA-1pt [5]	5.3×10^6 (2.8×10^6)	2.5×10^4 (1.5×10^4)
GA-2pt	4.0×10^6 (2.3×10^6)	1.6×10^4 (1.0×10^4)
LamarckiAnt		
$g = 0$, $m = 50$	1.5×10^8 (1.6×10^8)	2.7×10^5 (2.7×10^5)
$g = 0.25$, $m = 50$	1.3×10^8 (1.2×10^8)	1.9×10^5 (1.8×10^5)
$g = 0.5$, $m = 50$	8.3×10^7 (6.2×10^7)	1.3×10^5 (9.4×10^4)
$g = 0.75$, $m = 10$	4.7×10^7 (4.1×10^7)	7.8×10^4 (6.9×10^4)
$g = 0.75$, $m = 50$	4.8×10^7 (4.3×10^7)	8.1×10^4 (7.2×10^4)
$g = 0.75$, $m = 100$	5.2×10^7 (6.2×10^7)	8.9×10^4 (1.1×10^5)

Table 4: Mean first encounter times for 100 global optimisation runs from random starting positions of BLN-69. Values in parentheses are the standard deviations.

4 Conclusions

We have shown that the LamarckiAnt algorithm is competitive with the best available optimisation algorithms for a class of difficult global optimisation problems. For BLN-46, the mean number of minimisations before first encounter of the global minimum are lower than those found with BH or GA. The performance is less impressive for BLN-69, but it is close enough to the best available algorithms to suggest that further optimisation is worthwhile. The use of a restart operator combined with a global best update that allows information to be transferred between restarts gives a substantial improvement in the efficiency of LamarckiAnt for the larger system. In future work, we will implement the LamarckiAnt algorithm for all-atom models of molecules, such as the AMBER [33] and CHARMM [34] force fields, as well as other coarse-grained protein models.

5 Acknowledgements

The computations described in this paper were performed using the University of Birmingham's BlueBEAR HPC service, which provides a High Performance Computing service to the University's research community. See http://www.birmingham.ac.uk/bear for more details.

References

[1] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys*, vol. 21, no. 6, pp. 1087–1092, 1953. [Online]. Available: http://dx.doi.org/10.1063/1.1699114

[2] Z. Li and H. A. Scheraga, "Monte Carlo-minimization approach to the multiple-minima problem in protein folding," *Proc. Natl. Acad. Sci. USA*, vol. 84, no. 19, pp. 6611–6615, 1987. [Online]. Available: http://www.pnas.org/content/84/19/6611.abstract

[3] D. J. Wales and J. P. K. Doye, "Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms," *J. Phys. Chem. A*, vol. 101, no. 28, pp. 5111–5116, 1997. [Online]. Available: http://dx.doi.org/10.1021/jp970984n

[4] D. J. Wales and H. A. Scheraga, "Global Optimization of Clusters, Crystals, and Biomolecules," *Science*, vol. 285, no. 5432, pp. 1368–1372, 1999. [Online]. Available: http://dx.doi.org/10.1126/science.285.5432.1368

[5] M. T. Oakley, D. J. Wales, and R. L. Johnston, "Energy Landscape and Global Optimization for a Frustrated Model Protein," *J. Phys. Chem. B*, vol. 115, no. 39, pp. 11 525–11 529, 2011. [Online]. Available: http://dx.doi.org/10.1021/jp207246m

[6] ——, "The Effect of Nonnative Interactions on the Energy Landscapes of Frustrated Model Proteins," *Journal of Atomic, Molecular, and Optical Physics*, vol. 2012, p. 192613, 2012. [Online]. Available: http://dx.doi.org/10.1155/2012/192613

[7] D. M. Deaven, N. Tit, J. R. Morris, and K. M. Ho, "Structural optimization of Lennard-Jones clusters by a genetic algorithm," *Chem. Phys. Lett.*, vol. 256, no. 1-2, pp. 195–200, Jun. 1996. [Online]. Available: http://dx.doi.org/10.1016/0009-2614(96)00406-x

[8] R. L. Johnston, "Evolving better nanoparticles: Genetic algorithms for optimising cluster geometries," *Dalton Trans.*, no. 22, pp. 4193–4207, 2003. [Online]. Available: http://dx.doi.org/10.1039/b305686d

[9] R. Unger, "The Genetic Algorithm Approach to Protein Structure Prediction," in *Applications of Evolutionary Computation in Chemistry*, ser. Structure & Bonding, R. L. Johnston, Ed. Springer Berlin / Heidelberg, 2004, vol. 110, pp. 2697–2699. [Online]. Available: http://dx.doi.org/10.1007/b13936

[10] M. J. Vainio and M. S. Johnson, "Generating Conformer Ensembles Using a Multiobjective Genetic Algorithm," *J. Chem. Inf. Model.*, vol. 47, no. 6, pp. 2462–2474, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1021/ci6005646

[11] T. Huber and W. F. van Gunsteren, "SWARM-MD: Searching Conformational Space by Cooperative Molecular Dynamics," *J. Phys. Chem. A*, vol. 102, no. 29, pp. 5937–5943, Jun. 1998. [Online]. Available: http://dx.doi.org/10.1021/jp9806258

[12] J. Lv, Y. Wang, L. Zhu, and Y. Ma, "Particle-swarm structure prediction on clusters," *J. Chem. Phys*, vol. 137, no. 8, p. 084104, 2012. [Online]. Available: http://dx.doi.org/10.1063/1.4746757

[13] A. J. Bennett, R. L. Johnston, E. Turpin, and J. Q. He, "Analysis of an Immune Algorithm for Protein Structure Prediction," *Informatica*, vol. 32, no. 2, 2008.

[14] H. A. Bahamish, R. Abdullah, and R. A. Salam, "Protein Tertiary Structure Prediction Using Artificial Bee Colony Algorithm," *Third Asia International Conference on Modelling & Simulation*, pp. 258–263, 2009. [Online]. Available: http://dx.doi.org/10.1109/ams.2009.47

[15] A. Shmygelska and H. Hoos, "An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem," *BMC Bioinformatics*, vol. 6, no. 1, p. 30, 2005. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-6-30

[16] S. Fidanova and I. Lirkov, "Ant colony system approach for protein folding," *Proceedings of the International Multiconference on Computer Science and Information Technology*, pp. 887–891, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1109/IMCSIT.2008.4747347

[17] F. Daeyaert, M. De Jonge, L. Koymans, and M. Vinkers, "An ant algorithm for the conformational analysis of flexible molecules," *J. Comput. Chem.*, vol. 28, no. 5, pp. 890–898, 2007. [Online]. Available: http://dx.doi.org/10.1002/jcc.20595

[18] T. Dresselhaus, J. Yang, S. Kumbhar, and M. P. Waller, "A Hybrid Metaheuristic Approach for Non-Local Optimization of Molecular Systems," *J. Chem. Theory Comput.*, Feb. 2013. [Online]. Available: http://dx.doi.org/10.1021/ct301079m

[19] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, no. 1, pp. 29–41, Feb. 1996. [Online]. Available: http://dx.doi.org/10.1109/3477.484436

[20] P. G. Mezey, "Catchment region partitioning of energy hypersurfaces, I," *Theoretica Chimica Acta*, vol. 58, no. 4, pp. 309–330, 1981. [Online]. Available: http://dx.doi.org/10.1007/BF00553581

[21] J. D. Honeycutt and D. Thirumalai, "Metastability of the folded states of globular proteins," *Proc. Nat. Acad. Sci.*, vol. 87, no. 9, pp. 3526–3529, May 1990. [Online]. Available: http://www.pnas.org/cgi/content/abstract/87/9/3526

[22] R. S. Berry, N. Elmaci, J. P. Rose, and B. Vekhter, "Linking topography of its potential surface with the dynamics of folding of a protein model," *Proc. Natl. Acad. Sci. USA*, vol. 94, no. 18, pp. 9520–9524, 1997. [Online]. Available: http://www.pnas.org/cgi/content/abstract/94/18/9520

[23] M. A. Miller and D. J. Wales, "Energy landscape of a model protein," *J. Chem. Phys.*, vol. 111, no. 14, pp. 6610–6616, 1999. [Online]. Available: http://dx.doi.org/10.1063/1.480011

[24] D. J. Wales and P. E. J. Dewsbury, "Effect of salt bridges on the energy landscape of a model protein," *J. Chem. Phys.*, vol. 121, no. 20, pp. 10 284–10 290, 2004. [Online]. Available: http://dx.doi.org/10.1063/1.1810471

[25] S. A. Larrass, L. M. Pegram, H. L. Gordon, and S. M. Rothstein, "Efficient generation of low-energy folded states of a model protein. II. Automated histogram filtering," *J. Chem. Phys.*, vol. 119, no. 24, pp. 13 149–13 158, 2003. [Online]. Available: http://dx.doi.org/10.1063/1.1628671

[26] P. W. Pan, H. L. Gordon, and S. M. Rothstein, "Local-structural diversity and protein folding: Application to all-beta off-lattice protein models," *J. Chem. Phys.*, vol. 124, no. 2, p. 024905, 2006. [Online]. Available: http://dx.doi.org/10.1063/1.2151174

[27] J. Kim, J. E. Straub, and T. Keyes, "Statistical temperature molecular dynamics: Application to coarse-grained beta-barrel-forming protein

models," *J. Chem. Phys.*, vol. 126, no. 13, p. 135101, 2007. [Online]. Available: http://dx.doi.org/10.1063/1.2711812

[28] S. Y. Kim, "An off-lattice frustrated model protein with a six-stranded beta-barrel structure," *J. Chem. Phys.*, vol. 133, no. 13, p. 135102, 2010. [Online]. Available: http://dx.doi.org/10.1063/1.3494038

[29] D. A. Evans and D. J. Wales, "Free energy landscapes of model peptides and proteins," *J. Chem. Phys.*, vol. 118, no. 8, pp. 3891–3897, 2003. [Online]. Available: http://dx.doi.org/10.1063/1.1540099

[30] D. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1-3, pp. 503–528, Aug. 1989. [Online]. Available: http://dx.doi.org/10.1007/BF01589116

[31] D. J. Wales, "GMIN: A program for finding global minima and calculating thermodynamic properties from basin-sampling." [Online]. Available: http://www-wales.ch.cam.ac.uk/GMIN/

[32] T. Stützle and H. H. Hoos, "MAX-MIN Ant System," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, Jun. 2000. [Online]. Available: http://dx.doi.org/10.1016/s0167-739x(00)00043-1

[33] R. Salomon-Ferrer, D. A. Case, and R. C. Walker, "An overview of the Amber biomolecular simulation package," *WIREs Comput Mol Sci*, vol. 3, no. 2, pp. 198–210, Mar. 2013. [Online]. Available: http://dx.doi.org/10.1002/wcms.1121

[34] B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor,

R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus, "CHARMM: The biomolecular simulation program," *J. Comput. Chem.*, vol. 30, no. 10, pp. 1545–1614, Jul. 2009. [Online]. Available: http://dx.doi.org/10.1002/jcc.21287